NTNU – Trondheim
Norwegian University of
Science and Technology

Network Layer — Control Plane

TTM4200 – Computer Networks

individual Readiness Assurance Test

- Find the right answer **alone.**
- Choose the answer that fits **best.**
- **One** handwritten A4 page as helping material.

20:00

team Readiness Assurance Test

- Find the right answer **in teams**.
- Choose the answer that fits **best.**
- **One** handwritten A4 page as helping material.

20:00

Time for a break

we restart at 11:15

Join at menti.com | use code 1292 1024

NTNU
Kunnskap for en bedre verden

Instructions

Go to

www.menti.com

Enter the code

1292 1024

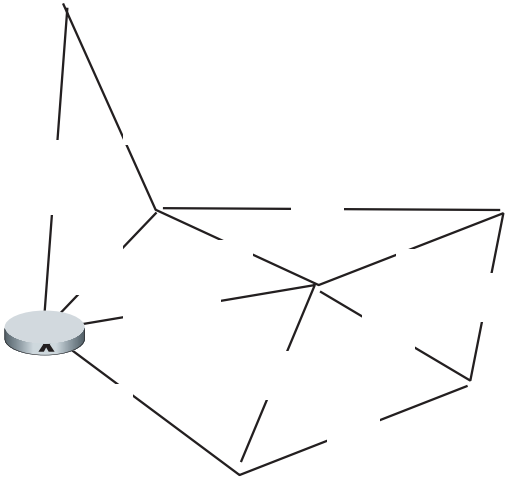
Or use QR code

Problem 3

- Groups are routers
- Routers communicate only with neigbours
  - Post-it send via group member
  - Unidirectional communication → Make sure to keep your neighbors postet!
- Information is forwarded unaltered → no information aggregation
- Build the topology ;)
  - Find neighbors
  - Inform others about your local information
  - Help to inform everyone about others' local information

Topology?!

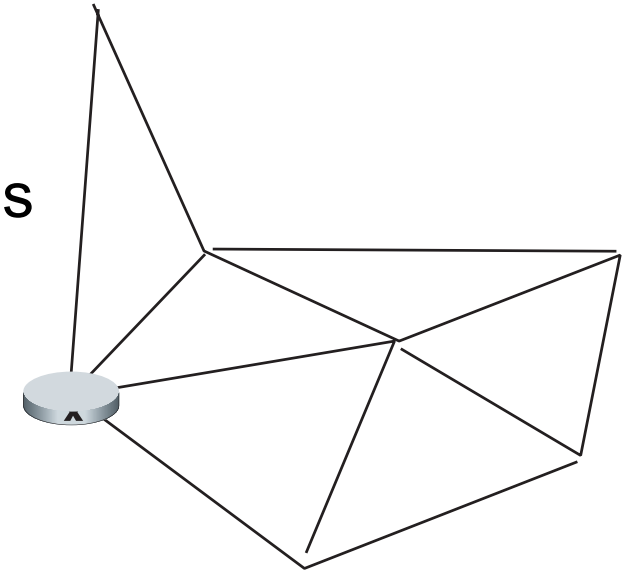
Solving Problem 3
(pg. 469)



step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)

Problem 3 (*pg. 469*)

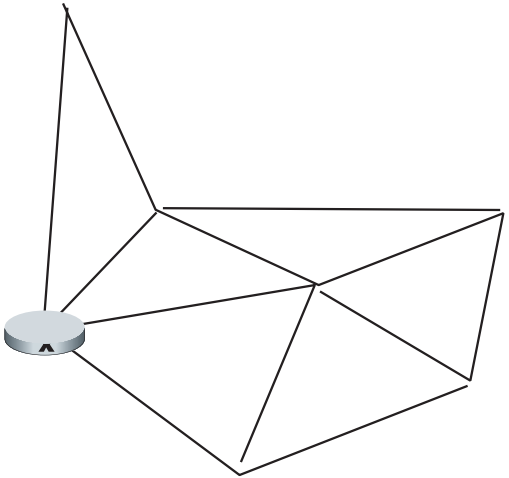
Consider the following network:
With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from *x* to all network nodes. Show how the algorithm works by computing a table similar to the table below.



step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)

10m

Solving Problem 3
(pg. 469)



step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)

Welcome

We start at 10:15

Remember that the RATs can include anything that has already been covered in class!

RATs

Question 2: A global routing algorithm computes the least-cost path between a source and destination:

by using complete, global knowledge about the network in each router or in a centralised control plane. (This is the best answer.)

48.8%

in an iterative manner where no node has the complete information about the costs of all network links.

2.4%

by using complete, global knowledge about the network in a centralised data plane.

14.6%

in an iterative manner where a central control plane has the complete information about the costs of all network links.

34.1%

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing algorithm classification

global: all routers have *complete* topology, link cost info

- "link state" algorithms

How fast do routes change?

static: routes change slowly over time

dynamic: routes change more quickly

- periodic updates or in response to link cost changes

decentralized: iterative process of computation, exchange of info with neighbors

- routers initially only know link costs to attached neighbors
- "distance vector" algorithms

global or decentralized information?

RATs

Question 6: Comparing link-state (LS) with distance-vector (DV) algorithms we can say that:

LS typically has a higher message complexity. (This is the best answer.)

36.6%

DV converges faster.

12.2%

DV is more robust to erroneous routing information.

24.4%

LS typically has lower message complexity.

26.8%

RATs

Question 7: Assume a network with three nodes: k, l and m. Suppose the initial link costs are $c(k,l)=5$, $c(k,m)=20$ and $c(l,m)=3$ and that a distance-vector algorithm computes all the least-cost paths. If suddenly the cost of link (k,l) increases from 5 to 40:

we will have a transient routing loop between m and l. (This is the best answer.)

36.6%

we will have a transient routing loop between k and l.

19.5%

a new route from l through m to k will be immediately calculated.

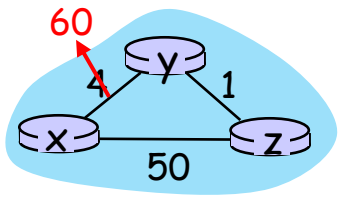
24.4%

a new route from m to k via the direct link will be immediately calculated.

19.5%

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Distance vector: link cost changes



link cost changes:

- node detects local link cost change
- "bad news travel slow" – count-to-infinity problem:
  - *y* sees direct link to *x* has new cost 60, but z has said it has a path at cost of 5. So *y* computes "my new cost to x will be 6, via z); notifies *z* of new cost of 6 to *x.*
  - *z* learns that path to *x* via *y* has new cost 6, so *z* computes "my new cost to *x* will be 7 via y), notifies *y* of new cost of 7 to *x.*
  - *y* learns that path to *x* via *z* has new cost 7, so *y* computes "my new cost to *x* will be 8 via y), notifies *z* of new cost of 8 to *x.*
  - *z* learns that path to *x* via *y* has new cost 8, so *z* computes "my new cost to *x* will be 9 via y), notifies *y* of new cost of 9 to *x.*

  …
- see text for solutions. *Distributed algorithms are tricky!*

RATs

Question 8: Routing loops can result in:

increased packet loss if the TTL of an IP datagram is reached. (This is the best answer.)

53.7%

increased delay until the loop is removed but not in losses.

22.0%

increased routing information being sent by LS routing protocols.

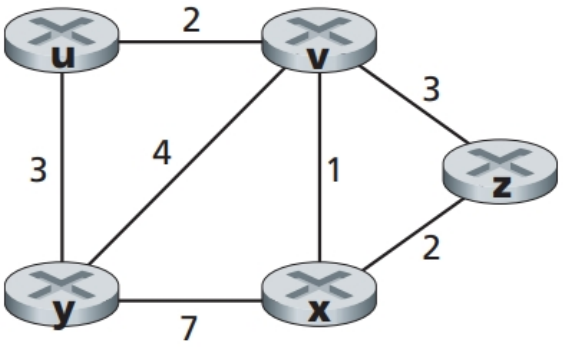
0.0%

increased routing information being sent by DV routing protocols.

24.4%

Problem 5 (*pg. 470*) (not solved in class)

Consider the network shown below. Assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node *z*. (**interactive!**)



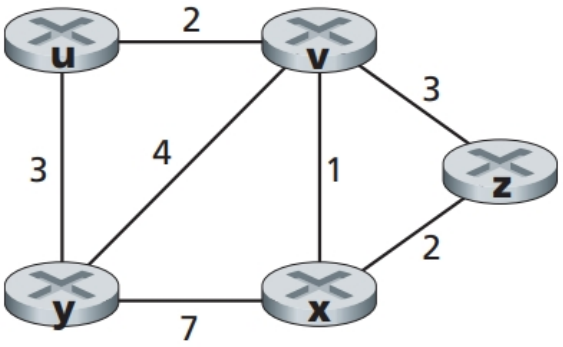
Cost to

from	u	v	x	y	z
v					
x					
z					

10m

Problem 5 *(pg. 470)*

Consider the network shown below. Assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node *z*. (**interactive!**)



Cost to

from	u	v	x	y	z
v					
x					
z					

At node *z*

Cost to

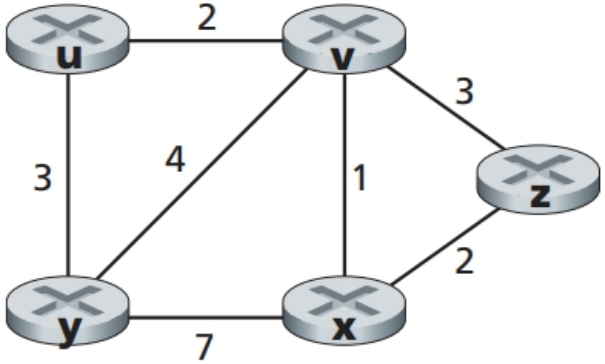
from	u	v	x	y	z
v					
x					
z					

Cost to

from	u	v	x	y	z
v					
x					
z					

Cost to

from	u	v	x	y	z
v					
x					
z					

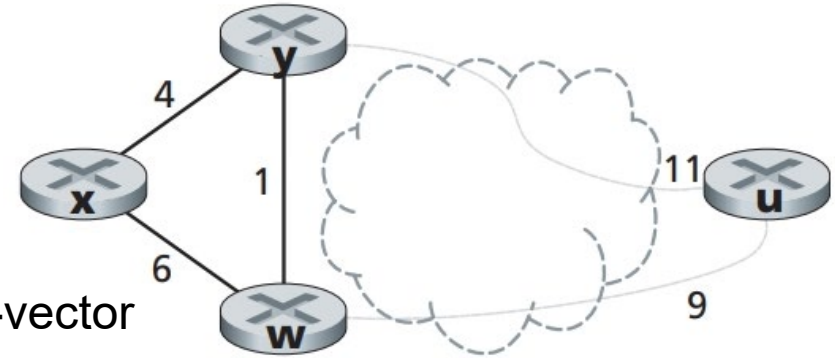


Problem 7 *(pg. 470)*

Consider the network fragment shown below. x has only two attached neighbours, w and y. *w* has a minimum-cost path to destination u (dotted line network) of 9, and y has a minimum-cost path to u of 11. The complete paths from w and y to u (and between w and y) are pictured with dotted lines, as they are irrelevant to the solution.

a) Give x's distance vector for destinations w, y and u.

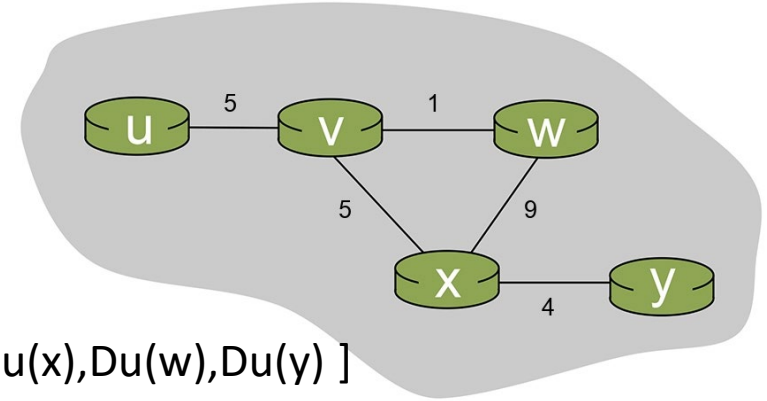
b) Give a link cost-change for either $c(x, w)$ or $c(x, y)$ such that x will inform its neighbours of a new minimum-cost path to u as a result of executing the distance-vector algorithm



10m

Extra DV (not solved in class)

- When the algorithm converges, what are the distance vectors in each of the routers. Note that this network is simple enough that you should be able to determine the final distance vectors by inspection (i.e., without having to execute the distance vector algorithm).
- What are the initial distance vectors, when each node only knows the one-hop link cost to its directly-attached neighbors?

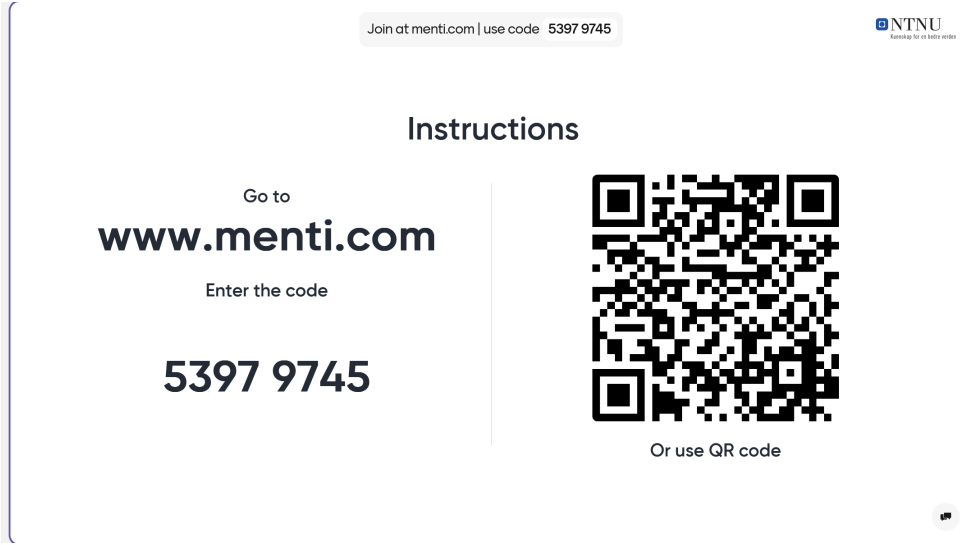


e.g.: Du = [Du(u),Du(v),Du(x),Du(w),Du(y)]

10m

Break

- We continue 11:15



Readings for next week

Book pages

- 480-484 (full Sec. 6.1)
- 507-515 (6.4 – 6.4.2)
- 521-527 (6.4.3)

- Optional:
  - 484-505 (Sec. 6.2, 6.3 1-3)
  - 542–547 (6.7 highly recommended)

Key concepts

MAC/Phy addresses

ARP/ARP table

Filtering vs. Switching

Self-learning switches

Switch vs. router

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer and LANs: our goals

- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet, VLANs
- datacenter networks

- instantiation, implementation of various link layer technologies

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

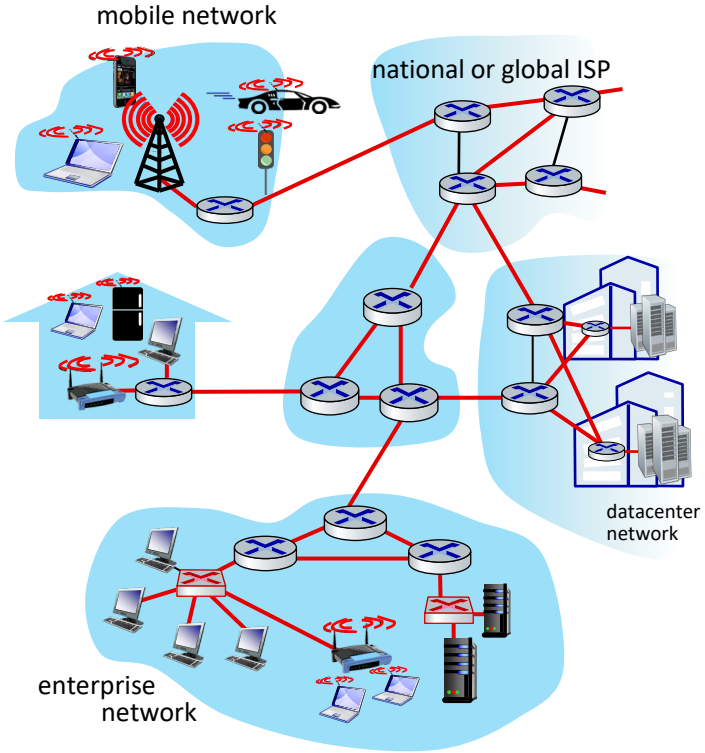
- a day in the life of a web request

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram



link layer has responsibility of transferring datagrams from one node to a physically adjacent node over a link

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer: context

- datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link
- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

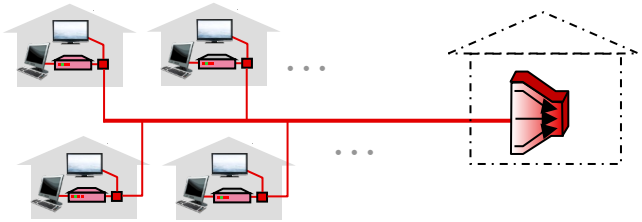
transportation analogy:

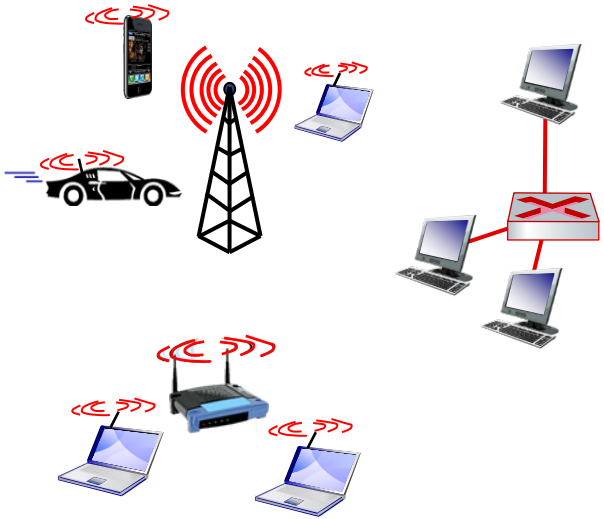
- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link-layer protocol
- travel agent = routing algorithm

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer: services

- framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses in frame headers identify source, destination (different from IP address!)
- reliable delivery between adjacent nodes
  - we already know how to do this!
  - seldom used on low bit-error links
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

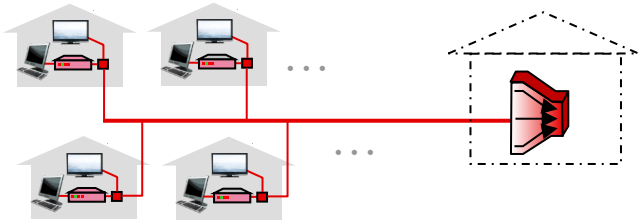


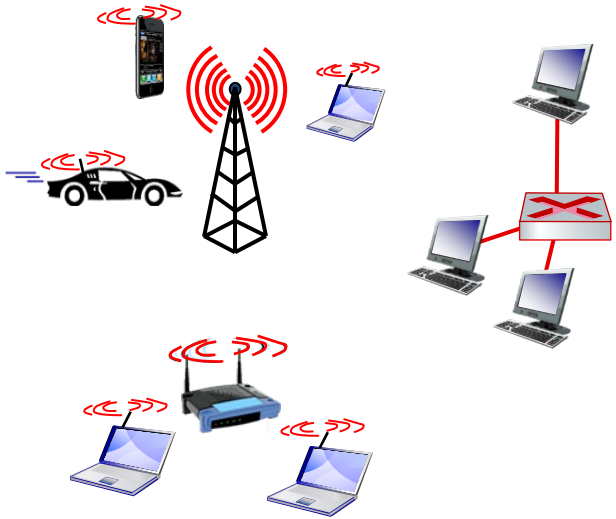


Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer: services (more)

- flow control:
  - pacing between adjacent sending and receiving nodes
- error detection:
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame
- error correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- half-duplex and full-duplex:
  - with half duplex, nodes at both ends of link can transmit, but not at same time

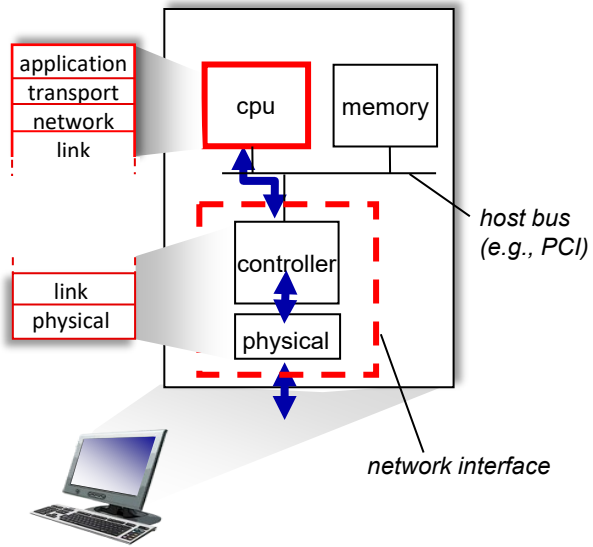




Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

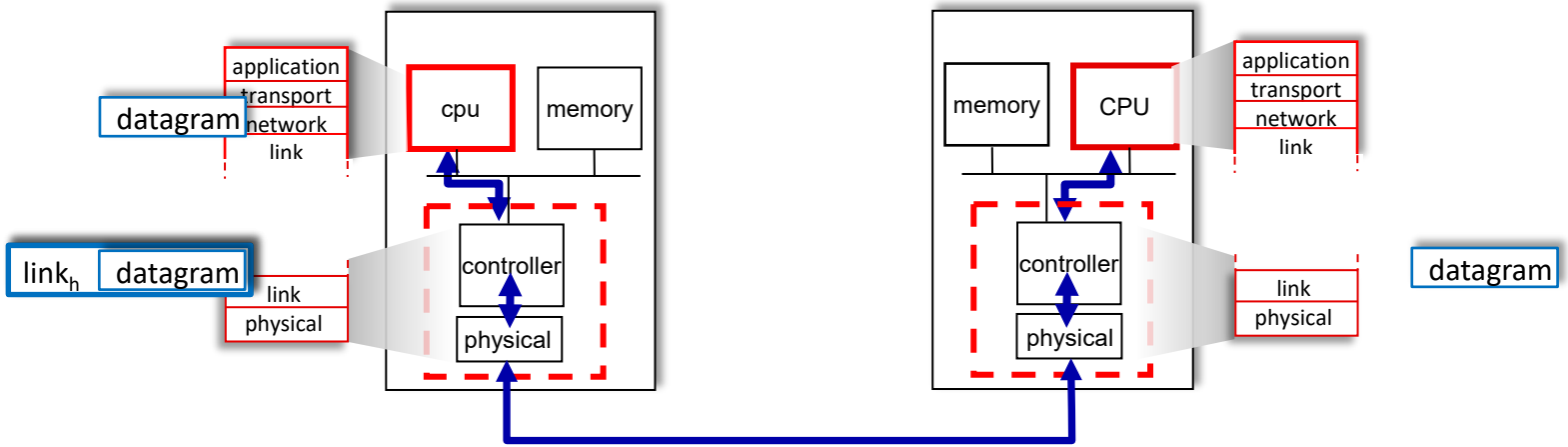
Where is the link layer implemented?

- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Interfaces communicating



sending side:

- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer, LANs: roadmap

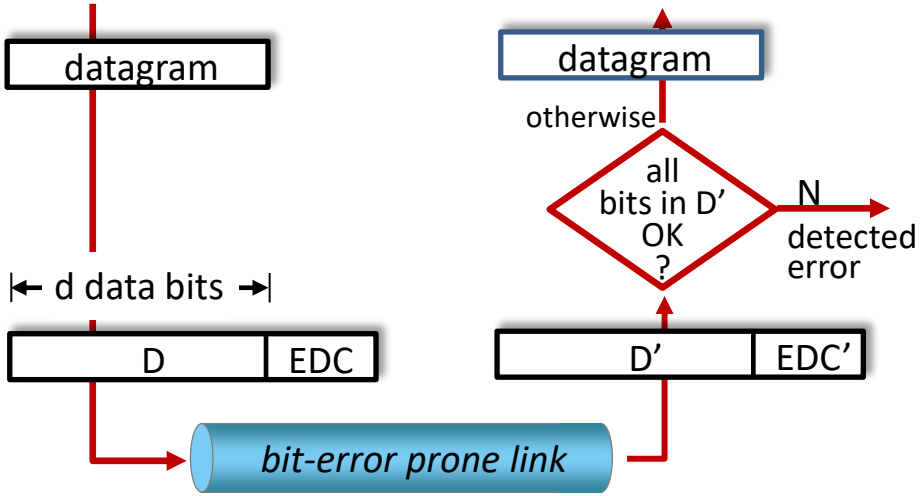
- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Error detection

EDC: **e**rror **d**etection and **c**orrection bits (e.g., redundancy)
D: data protected by error checking, may include header fields



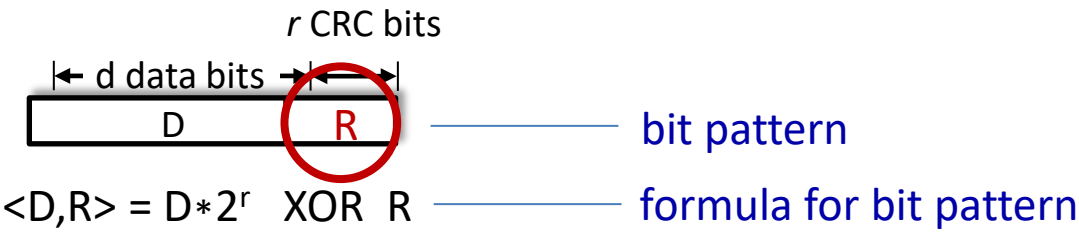
Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Cyclic Redundancy Check (CRC)

- more powerful error-detection coding
- D: data bits (given, think of these as a binary number)
- G: bit pattern (generator), of *r+1* bits (given)



goal: choose *r* CRC bits, R, such that <D,R> exactly divisible by G (mod 2)

- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi)

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Multiple access links, protocols

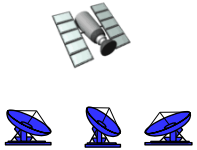
two types of "links":

- point-to-point
  - point-to-point link between Ethernet switch, host
  - PPP for dial-up access
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/5G, satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: satellite

humans at a cocktail party (shared air, acoustical)

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

An ideal multiple access protocol

given: **m**ultiple **a**ccess **c**hannel (MAC) of rate *R* bps

desiderata:

1. when one node wants to transmit, it can send at rate *R*.
2. when M nodes want to transmit, each can send at average rate *R/M*
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

MAC protocols: taxonomy

three broad classes:

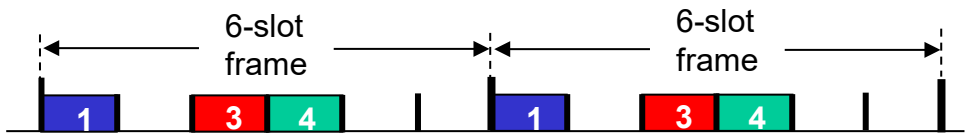
- channel partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- *random access*
  - channel not divided, allow collisions
  - "recover" from collisions
- "taking turns"
  - nodes take turns, but nodes with more to send can take longer turns

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle

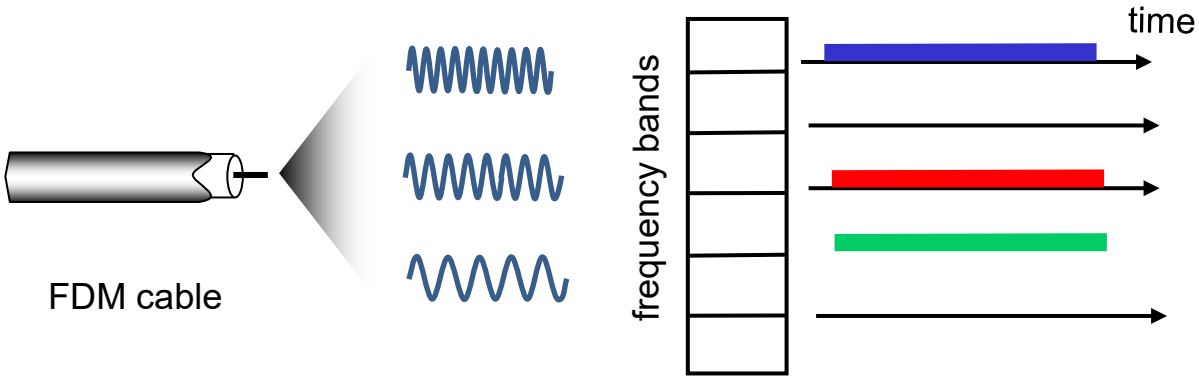


Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Random access protocols

- when node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes: "collision"
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission

- human analogy: don't interrupt others!

CSMA/CD: CSMA with *collision detection*

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless

- human analogy: the polite conversationalist

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

"Taking turns" MAC protocols

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols

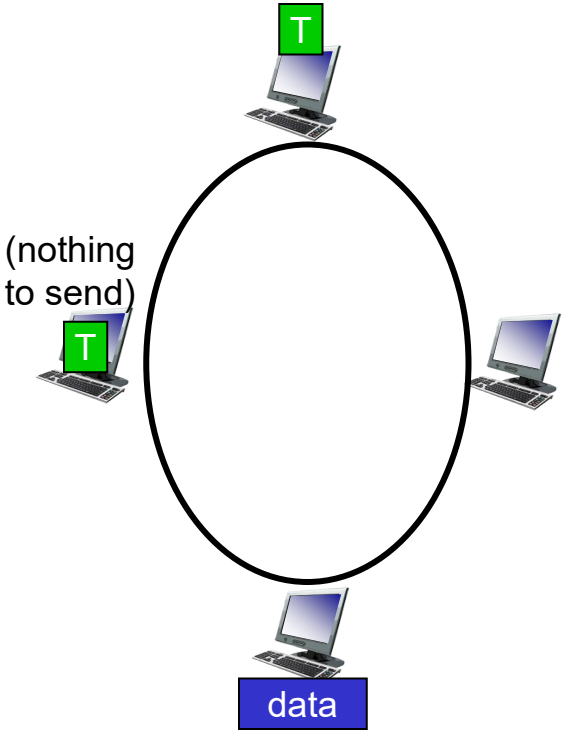
- look for best of both worlds!

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

"Taking turns" MAC protocols

token passing:

- control *token* passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Summary of MAC protocols

- channel partitioning, by time, frequency or code
  - Time Division, Frequency Division
- random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- taking turns
  - polling from central site, token passing
  - Bluetooth, FDDI, token ring

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

MAC addresses

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

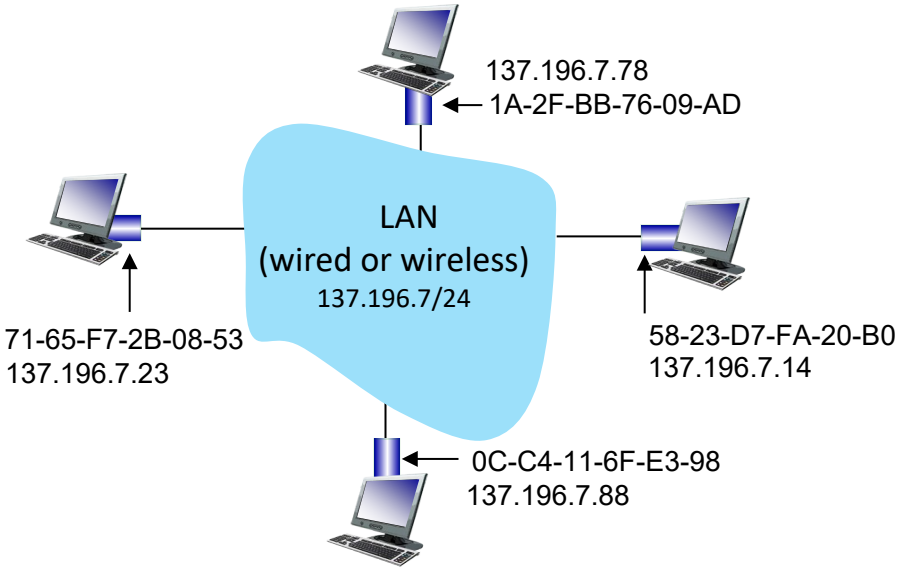
hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

MAC addresses

each interface on LAN

- has unique 48-bit MAC address
- has a locally unique 32-bit IP address (as we've seen)



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

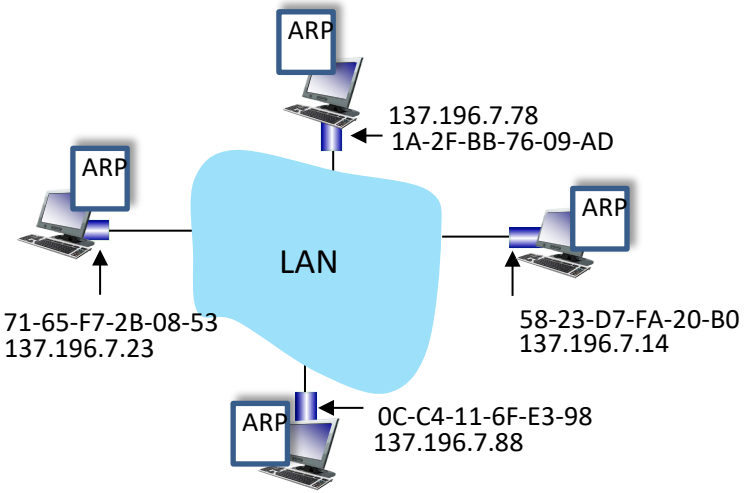
MAC addresses

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address: portability
  - can move interface from one LAN to another
  - recall IP address *not* portable: depends on IP subnet to which node is attached

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

ARP: address resolution protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

ARP protocol in action

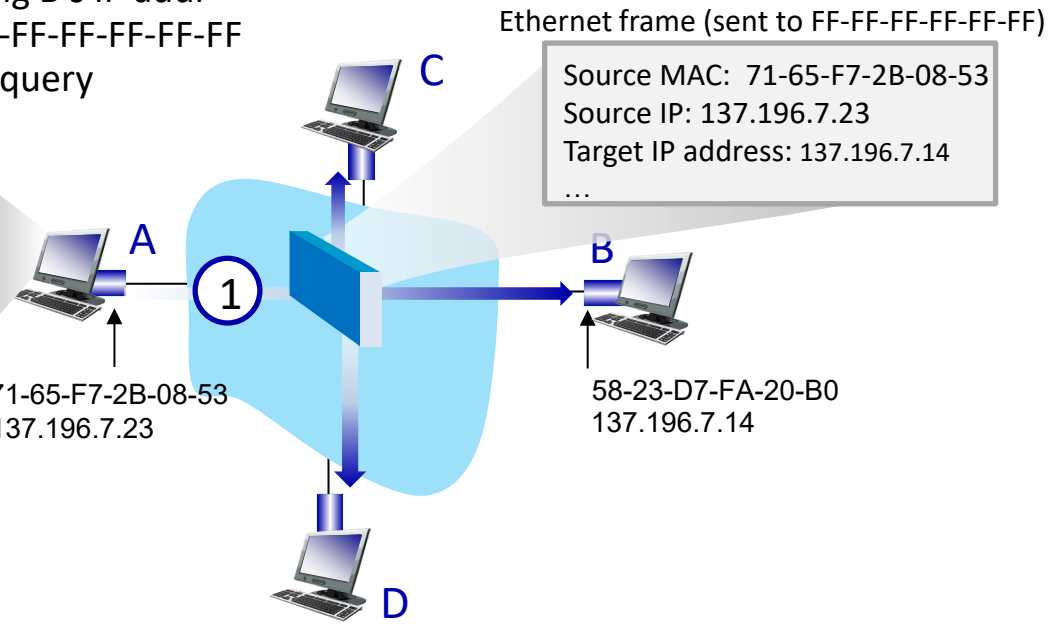
example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

1 A broadcasts ARP query, containing B's IP addr
- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

IP addr	MAC addr	TTL



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

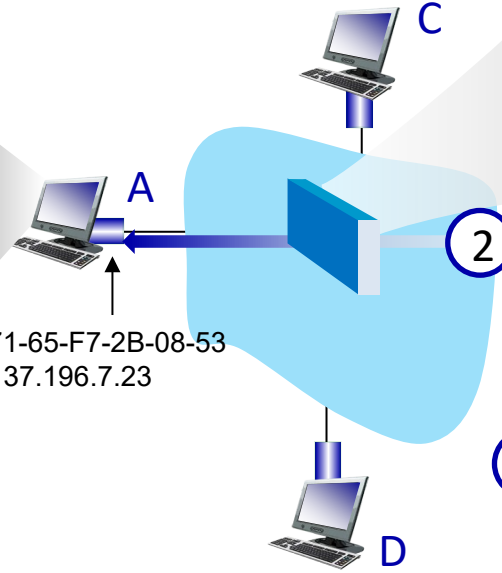
ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL



ARP message into Ethernet frame
(sent to 71-65-F7-2B-08-53)

Target IP address: 137.196.7.14
Target MAC address:
58-23-D7-FA-20-B0
...

2 B replies to A with ARP response, giving its MAC address

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

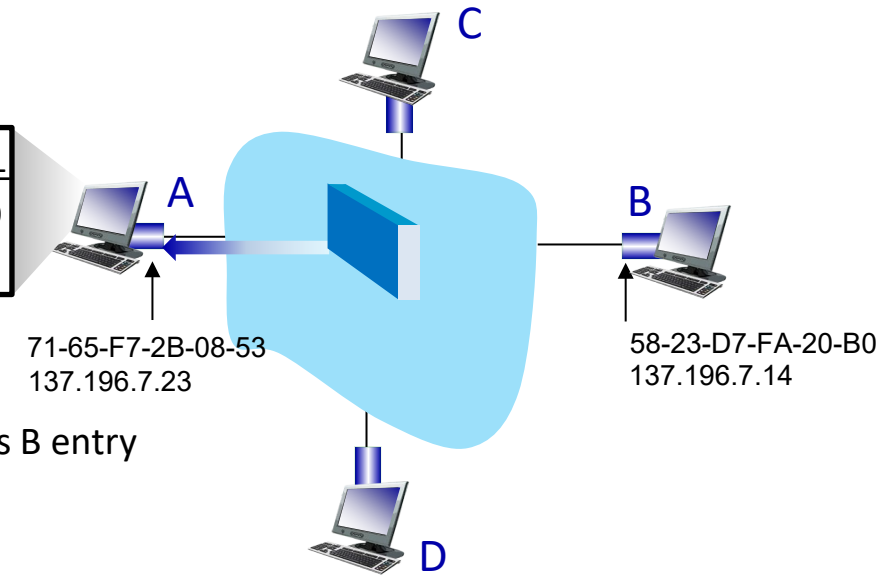
ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500



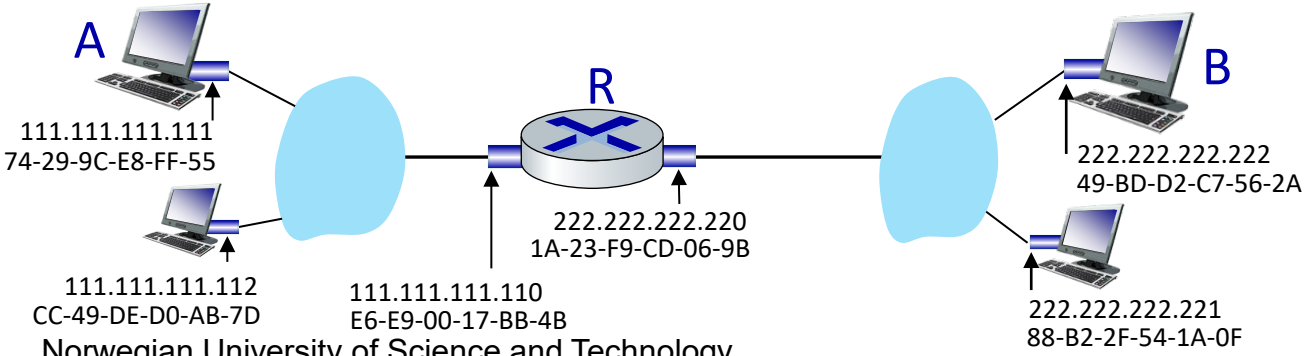
3 A receives B's reply, adds B entry into its local ARP table

Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

walkthrough: sending a datagram from *A* to *B* via *R*

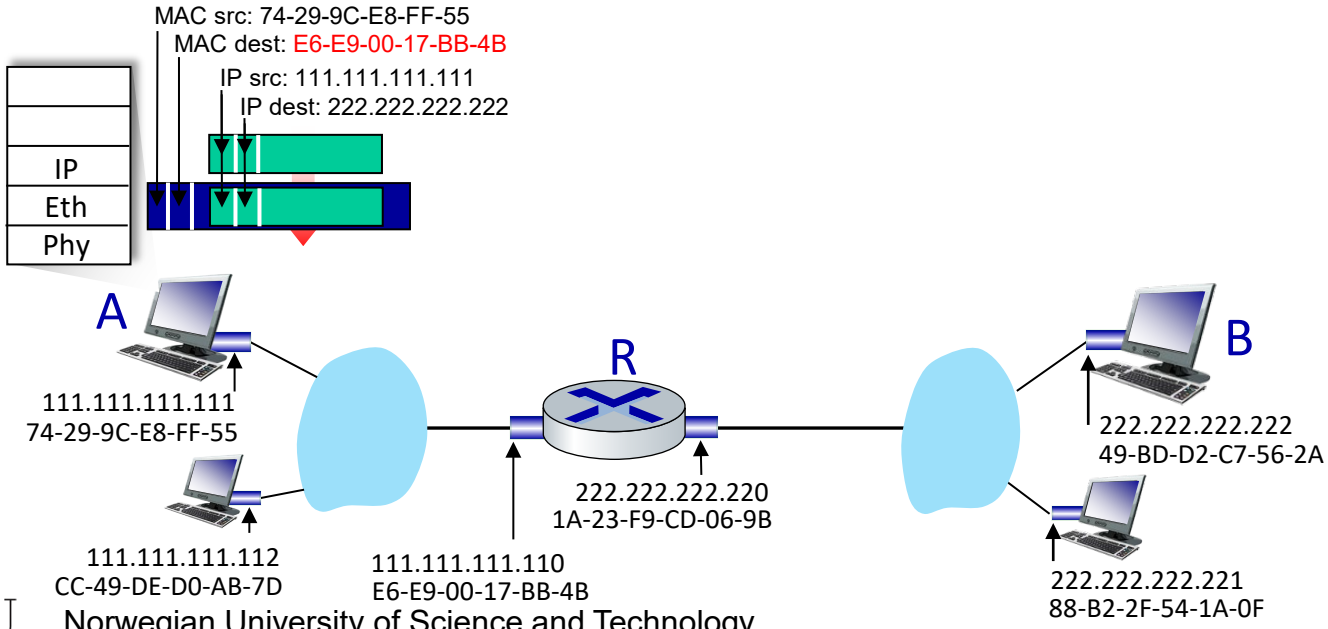
- focus on addressing – at IP (datagram) and MAC layer (frame) levels
- assume that:
  - A knows B's IP address
  - A knows IP address of first hop router, R (how?)
  - A knows R's MAC address (how?)



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

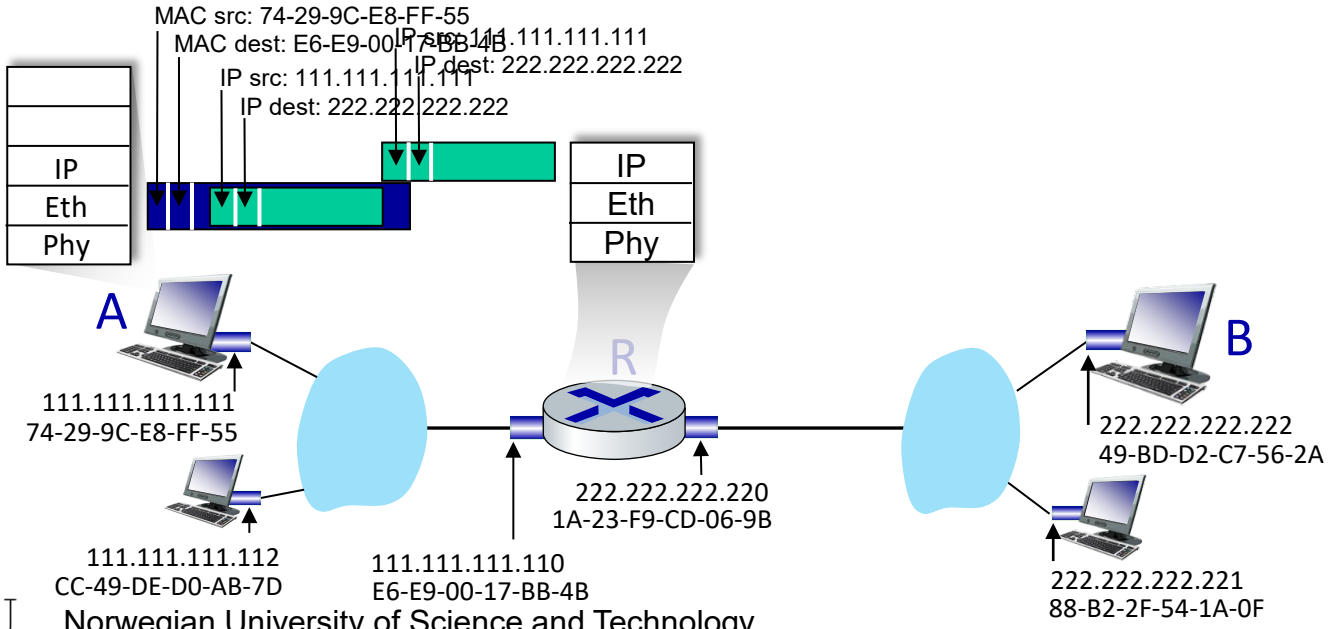
- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
  - R's MAC address is frame's destination



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

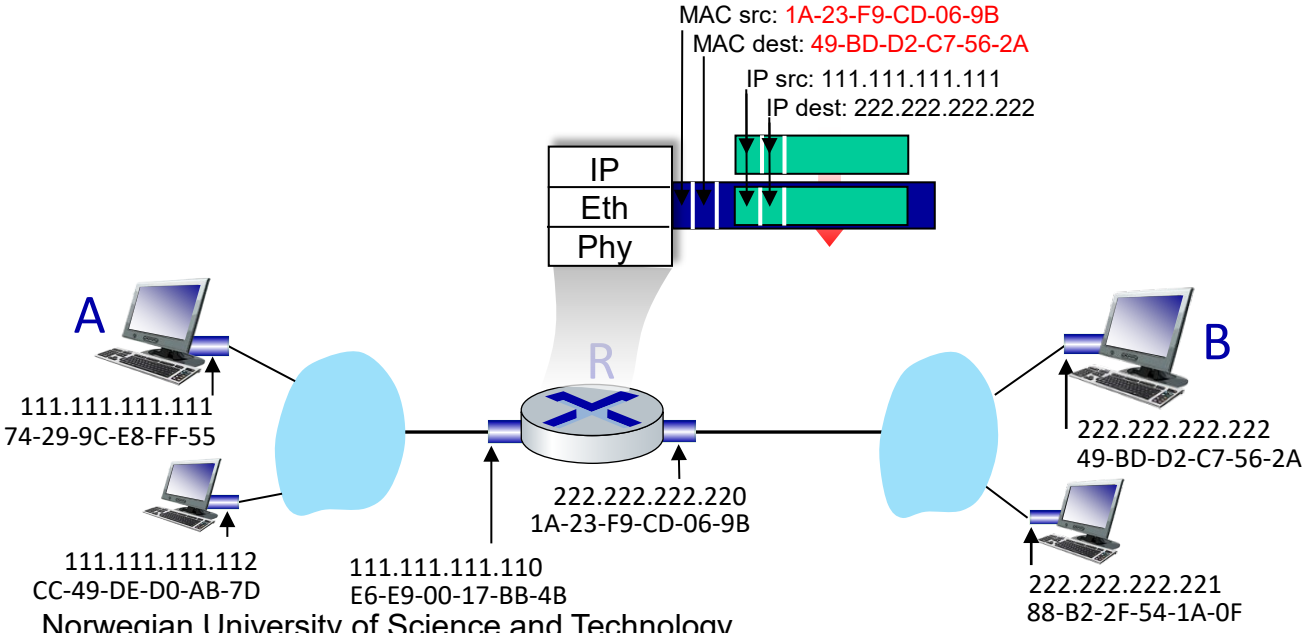
- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

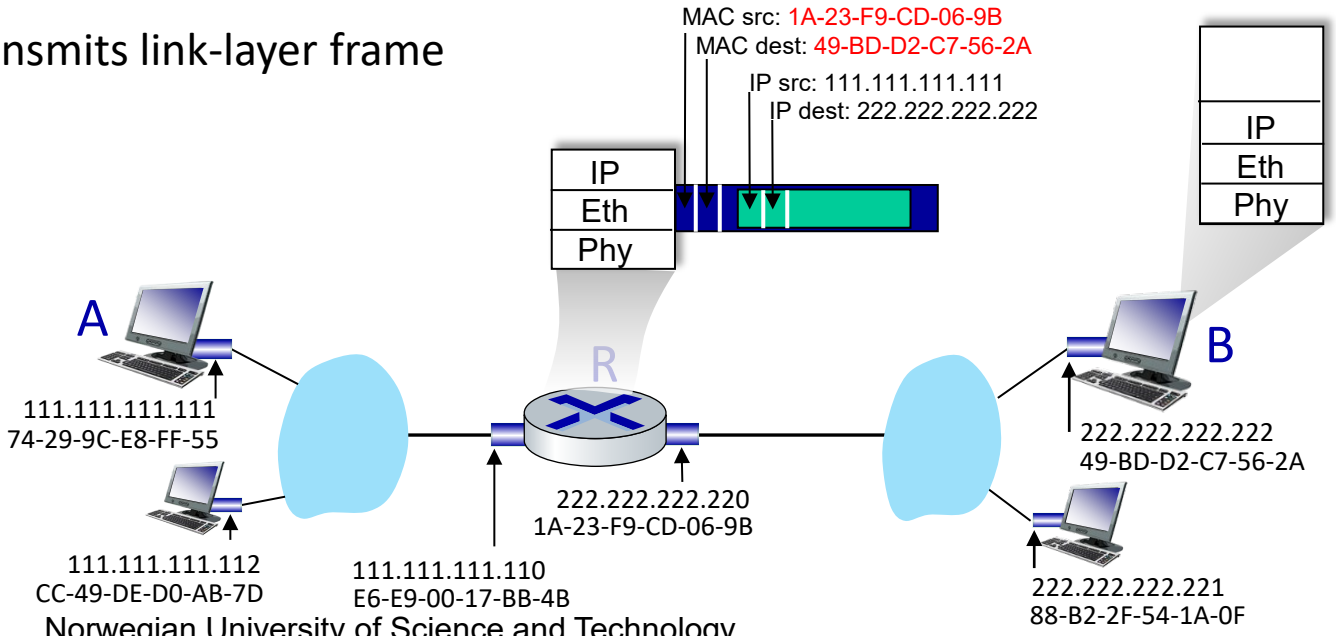
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

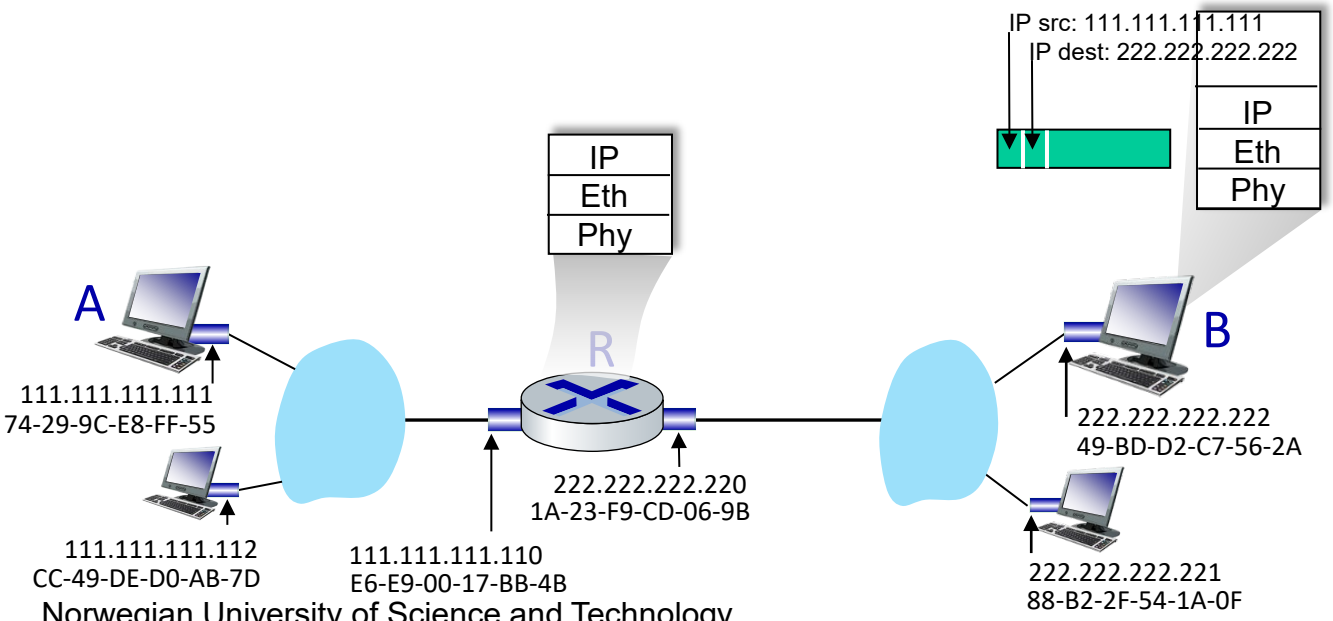
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame



Adapted from "*Computer Networking: A Top-Down Approach 8th edition*"
© Copyright 1996-2020 J.F Kurose and K.W. Ross, All Rights Reserved

Routing to another subnet: addressing

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



Readings for next week

Book pages

- 480-484 (full Sec. 6.1)
- 507-515 (6.4 – 6.4.2)
- 521-527 (6.4.3)

- Optional:
  - 484-505 (Sec. 6.2, 6.3 1-3)
  - 542–547 (6.7 highly recommended)

Key concepts

MAC/Phy addresses

ARP/ARP table

Filtering vs. Switching

Self-learning switches

Switch vs. router